Software Design Document

<Project Name>

Student Names

Table of Contents

[1.0 System Vision 3](#_Toc46748622)

[1.1 Problem Background 3](#_Toc46748623)

[1.2 System Overview 3](#_Toc46748624)

[1.3 Potential Benefits 3](#_Toc46748625)

[2.0 Requirements 4](#_Toc46748626)

[2.1 User Requirements 4](#_Toc46748627)

[2.2 Software Requirements 4](#_Toc46748628)

[2.3 Use Cases 4](#_Toc46748629)

[3.0 System Components and Software Design 5](#_Toc46748630)

[3.1 System Components 5](#_Toc46748631)

[3.2 Software Design 5](#_Toc46748632)

[4.0 User Interface Design 6](#_Toc46748633)

# System Vision

## Problem Background

With the rapid urbanization and growth of Victoria State, the road network has seen an increase in vehicular traffic. This growth, while indicative of economic and infrastructural development, has also led to a rise in traffic accidents. Local authorities, urban planners, and various stakeholders are in dire need of a tool that can help them understand the patterns, causes, and locations of these accidents. The Victoria State Accident Dataset, while comprehensive, is vast and complex, making it challenging to derive actionable insights without specialized tools.

## System Overview

The proposed software is a data analysis and visualization platform tailored specifically for the Victoria State Accident Dataset. It is designed to simplify the process of data interpretation, allowing users to perform a variety of analysis tasks, visualize data trends, and extract meaningful insights. The software offers an intuitive graphical interface, robust filtering tools, a range of visualization options, and predictive analysis capabilities. It aims to bridge the gap between raw data and actionable insights, providing stakeholders with the information they need to make informed decisions.

## Potential Benefits

Potential Benefits

1. **Enhanced Understanding**: The software aids in visualizing data, making patterns and trends more discernible, which can lead to a deeper understanding of accident causes and prevalent areas.
2. **Data-Driven Decisions**: Local authorities and urban planners can use the insights derived from the software to make informed decisions, potentially improving road safety.
3. **Informed Decision Making**: By visualizing accident hotspots and trends, local authorities can prioritize areas for infrastructure improvements, enhancing road safety.
4. **Efficient Resource Allocation**: With clear insights into accident patterns, resources like emergency services, road maintenance crews, and safety campaigns can be directed where they are needed the most.
5. **Proactive Planning**: The software's predictive analysis features allow urban planners and traffic engineers to anticipate future challenges and plan accordingly, potentially preventing accidents before they happen.
6. **Stakeholder Collaboration**: The software facilitates collaborative analysis, enabling various stakeholders, from traffic authorities to insurance professionals, to work together, fostering a holistic approach to road safety.
7. **Public Awareness**: By understanding the primary causes and locations of accidents, targeted public awareness campaigns can be launched, educating the public and promoting safer driving habits.

# Requirements

## User Requirements

**Traffic Engineers' Perspective**:

Traffic Engineers are at the forefront of ensuring the safety and efficiency of road networks. Their roles encompass the planning, design, and management of traffic systems, and they often collaborate with urban planners, civil engineers, and local authorities. Given the critical nature of their work, the software tailored to the Victoria State Accident Dataset should cater to their specific needs.

* **Intuitive Data Access**: Traffic Engineers should be able to effortlessly upload and access the dataset in its CSV format. The software should provide tools that allow them to easily navigate through the extensive data.
* **Focused Analysis Tools**: Given the vastness of the dataset, the software should offer robust filtering tools. Engineers should be able to narrow down data based on specific regions, types of accidents, or time frames, such as peak traffic hours.
* **Comprehensive Visualization**: The software must provide a range of visualization tools. Heatmaps to quickly identify accident hotspots, line graphs to track trends over time, and bar charts to compare accident frequencies across different regions or criteria are essential.
* **Predictive Capabilities**: As proactive planners, Traffic Engineers would benefit from any predictive features the software might offer. Being able to forecast potential future accident hotspots or trends based on current data would be invaluable.
* **Export and Reporting**: After their analysis, Traffic Engineers often need to present findings to various stakeholders, including city councils, planning committees, or the general public. The software should facilitate the easy export of visualizations, data summaries, and reports in commonly used formats.
* **Collaborative Features**: Given that Traffic Engineers often work in teams or collaborate with other departments, the software might offer features that support collaborative analysis, such as shared views, notes, or annotations.

## Software Requirements

* R1.1 The software shall provide an intuitive graphical user interface tailored for Traffic Engineers, ensuring ease of navigation and interaction.
* R1.2 The software shall support the direct uploading of the Victoria State Accident Dataset in its CSV format, ensuring compatibility with the specific data structure and columns.
* R1.3 The software shall offer robust data filtering tools, enabling Traffic Engineers to refine the dataset based on criteria such as specific regions, accident types, date ranges, or time frames.
* R1.4 The software shall provide a comprehensive suite of visualization tools, including heatmaps for accident hotspots, line graphs for trend analysis, and bar charts for comparative studies.
* R1.5 The software shall integrate predictive analysis capabilities, allowing Traffic Engineers to forecast potential future accident trends or hotspots based on the current dataset.
* R1.6 The software shall facilitate the export of visualizations, data summaries, and reports in popular formats such as PNG, JPEG, PDF, or even CSV for further analysis.
* R1.7 The software shall incorporate collaborative features, enabling multiple Traffic Engineers or stakeholders to work on shared views, add annotations, or leave notes on specific data points.
* R1.8 The software shall ensure data security and integrity, especially when multiple users are accessing or editing the dataset simultaneously.
* R1.9 The software shall provide a user manual or help section, offering guidance on its features, ensuring that Traffic Engineers can maximize the tool's capabilities.
* R1.10 The software shall be optimized for performance, ensuring quick data processing, visualization rendering, and minimal lag, given the potential size and complexity of the dataset.

## Use Cases & Use Case Diagrams

* + 1. **Use Case: Dataset Upload and Initialization**

**Description**: Traffic Engineers need to start their analysis by uploading the dataset.

**Steps**:

1. The user accesses the upload module.
2. The software prompts for the dataset file in CSV format.
3. The user selects the dataset, and the software validates and processes it.
4. Upon successful upload, the software initializes the dataset for analysis.
   * 1. **Use Case: Focused Data Filtering**

**Description**: Given the vastness of the dataset, engineers often need to narrow down their focus.

**Steps**:

1. The user interacts with the data filtering tools.
2. The software provides options to refine data based on regions, accident types, dates, etc.
3. The user sets desired filters, and the software updates the view accordingly.
   * 1. **Use Case: Visualization Creation and Analysis**

**Description**: Visual representations are crucial for Traffic Engineers to identify patterns and trends.

**Steps**:

1. The user selects the desired visualization tool (e.g., heatmap, line graph).
2. The software prompts the user for any necessary parameters or settings.
3. Upon confirmation, the software generates the visualization based on the selected data.
   * 1. **Use Case: Predictive Analysis for Future Planning**

**Description**: Engineers might want to forecast future accident trends.

**Steps**:

1. The user accesses the predictive analysis module.
2. The software offers options to set parameters for the prediction.
3. The user confirms, and the software provides a forecast based on historical data and trends.
   * 1. **Use Case: Exporting Visualizations and Reports**

**Description**: After analysis, findings often need to be shared with stakeholders.

**Steps**:

1. The user selects the export option.
2. Software prompts for the desired format and destination.
3. The user confirms, and the software exports the data or visualization as per the user's specifications.

# Software Design and System Components

## Software Design

A block diagram/flowchart of how your software might work

## System Components

### Functions

Preliminary list of all functions in the software. For each function in the list the following information is provided:

* a brief description of what it does (1 or 2 sentences);
* a list of the input parameters, and their data types, and what they are used for;
* a list of any side effects caused by the function (ie change global or member variables, changes data passed by reference from calling function etc)
* a description of the function’s return value

### Data Structures / Data Sources

List of all data structures in the software (eg linked lists, trees, arrays etc) or eternal data sources. For each data structure in the list the following information is provided:

* Type of structure (tree, list etc),
* Description of where and how it is used
* List of data members, and what each one is for do
* List of functions that use it

### Detailed Design

Pseudocode for all non-standard / non-trivial algorithms that operate on data structures

# User Interface Design

This is your initial interface design. Describe the tools you used for this design stage and any key findings that informed your design. This introduction is descriptive and should explain what you have completed for the actual design work you will present in the sub-sections below.

## Structural Design

Structural design refers to the navigational and information structure of your product – the structure that supports the interface layout. How will you structure your product? How will you group your information? How will you navigate through your product? Why? This can take the form of a diagram showing structure and hierarchy, supported by a discussion and justification of your choices. Why have you made these design choices? Describe and outline the structure of your interface and of your information.

## Visual Design

Detail your visual design: Layout, visual elements, icons, graphics, style, colour, fonts general screen designs. This can be sketches, wireframes, mockups etc, supported by a discussion, explanation, and justification of your choices.